INFORMATION AND COMPUTATION 91, 177–188 (1991)

Connectivity vs. Reachability

M. CHROBAK

Department of Mathematics, University of California, Riverside, California 92521

H. KARLOFF*

Department of Computer Science, University of Chicago, Chicago, Illinois 60637

AND

T. RADZIK†

Department of Computer Science, Stanford University, Stanford, California 94305-2140

We consider the problem of the relative complexity of the connectivity and reachability (s-t-connectivity) problems, in a model resembling the one used for graph properties. In our model an oracle answers queries about edge-induced subgraphs, and we count the number of queries made. The main result is that in order to determine whether t is reachable from s one has to ask $\Omega(n^2)$ questions about the connectivity of edge-induced subgraphs. For non-adaptive strategies we show that $\binom{n}{2}$ questions are necessary for $n \geqslant 6$. Several other results are included.

1. INTRODUCTION

This work has been motivated by the paper by Karchmer and Widgerson (1988), who proved that monotone circuits for reachability (often called also s-t-connectivity) must have depth $\Omega(\log^2 n)$. Though at first glance it seems almost obvious that the lower bound for graph connectivity should be the same, whether this is true was left in (Karchmer and Wigderson, 1988) as an open problem. Only very recently have Raz and Widgerson (1989) proven that the same lower bound holds for connectivity. However, this new proof does not use the lower bound for

* Research supported in part by NSF Grant CCR 8807534.
† Research supported in part by NSF Grant CCR 8858097.

0890-5401/91 $3.00

reachability and is in fact even more difficult. It appears that, despite the similarity between the two problems, there is no simple reduction of reachability to connectivity.

The aim of this paper is to shed some light on this difficulty, by considering the relative complexity of these problems, in an appropriately-defined oracle model of computation.

Let us define the problems we consider.

Connectivity. Given a graph G, determine if G is connected.

Reachability. Given a graph G and two vertices s and t, determine if there is a path from s to t.

We will usually that the vertices of G are numbered $1, 2, ..., n$, and that $s=1$, $t=n$.

We base our work on a model similar to that used in the area of graph complexity. Graph complexity deals with decision problems for graphs. Having such a decision problem, we investigate how many questions of the form "is (u, v) an edge of G?" have to be asked in order to determine whether a given graph has a desired property or not. (Throughout, both (u, v) and (v, u) will refer to the undirected edge between vertices u and v.) It is not hard to show, for example, that connectivity requires $\binom{n}{2}$ questions; we have to ask about all edges in the worst case. To prove this, we prove that in a game in which Player $\mathscr{A}$ asks questions and Player $\mathscr{B}$'s goal is to force $\mathscr{A}$ to query all edge slots, $\mathscr{B}$ has a winning strategy. (Every potential edge (u, v), for $1 \leqslant u \neq v \leqslant n$, is known as an *edge slot.*) The strategy is to answer "no" as long as possible. $\mathscr{B}$ starts with a complete graph G, and removes from G some edges during the game. Given a query about an edge (u, v), $\mathscr{B}$ checks if removing (u, v) disconnects G. If not, $\mathscr{B}$ removes (u, v) and answers "no." Otherwise, $\mathscr{B}$ leaves (u, v) in G and answers "yes." It is easy to see that all edges must be queried when $\mathscr{B}$ applies this strategy.

Properties like connectivity which require $\binom{n}{2}$ questions are called *evasive.* It is not hard to prove that reachability is evasive, too.

We study *relative graph complexity* by asking how difficult it is to determine reachability in a graph, given an oracle for connectivity, and vice versa. We consider both adaptive and non-adaptive strategies. An *adaptive* strategy is one in which the choice of $\mathscr{A}$'s queries can depend on previous answers. In a *non-adaptive* strategy, A simply gives B a collection of questions at once and uses the answers to determine whether or not a given property holds. Of course, adaptive strategies are more difficult to analyze.

Hajnal, Maass, and Turán (1988) (see also Lovaśz and Saks, 1988) investigated the communication commmplexity of graph properties and used their results about communication complexity to prove lower bounds for graph properties in a model in which the oracle answers queries of the form "does the set X of edge slots contain at least one edge of G?" In their model, both connectivity and reachability have complexity $\Theta(n \log n)$. For non-adaptive strategies they proved an $\Omega(n^2/\log^2 n)$ lower bound for the connectivity problem. Their model can also be viewed in terms of relative graph complexity, though the queries are fairly simple: about non-emptiness of a subgraph. In the related work of Aggarwal, Coppersmith, and Kleitman (1989), the oracle, given a set X of vertices, returns the number of edges in the subgraph induced by X. They showed that to reconstruct a graph $\Theta(n^2/\log n)$ queries are necessary and sufficient.

Our main result is that in order to determine whether t is reachable from s in a graph G, one has to ask $\Omega(n^2)$ queries about the connectivity of edge-induced subgraphs of G. We also improve slightly the trivial upper bound for this problem.

Another lower bound concerns non-adaptive strategies. We prove that for $n \geqslant 6$, exactly $\binom{n}{2}$ queries are necessary and sufficient to determine reachability given an oracle for connectivity, and thus we could instead query all edges. (For $n \leqslant 5$ it is possible to do a little better.)

2. Reducing Connectivity to Reachability

The results in this section are quite easy and are included only for completeness. Let $G = (V, E_G)$ be a graph. We will often fix $V = \{1, 2, ..., n\}$, while the set of edges E_G is unknown. If $U \subseteq V$, then by $E(U)$ we denote all edge slots in U, that is, the set $\{(u, v) \mid u, v \in U, u < v\}$. $E(v_1, ..., v_k)$ will be synonymous with $E(\{v_1, ..., v_k\})$. For a set $X \subseteq E(V)$ of edge slots, by $G|_X$ we denote the subgraph of G induced by those edges of G that are in X. In other words, the vertices of $G|_X$ are the endpoints of the edge slots in X, and the set of edges is $X \cap E_G$. We allow only questions of the form

(*) Given a nonempty set $X \subseteq E(V)$ and vertices s, t of $G|_X$ ($s \neq t$), is there a path from s to t in $G|_X$?

THEOREM 2.1. *$\mathcal{A}$ has a strategy which makes $n-1$ queries of the form (*), in the worst case, to determine if G is connected, and no strategy with fewer.*

Proof. $\mathcal{A}$'s strategy is to ask all questions (*) with $X = E(V)$, $s = 1$ and $t = 2, ..., n$. If the answers are all "yes" then G is connected. Otherwise it is not.

Now consider the lower bound. Let the first question of $\mathcal{A}$ be (X, s, t). $\mathcal{B}$ decides that the only edge incident to t is (s, t) and answers accordingly.

Now $\mathscr{A}$ has to determine whether the subgraph G' of G induced by $V-\{t\}$ is connected having no information about the edges of G'. An inductive argument shows that $\mathscr{A}$ must ask at least $n-1$ queries in total. ∎

Note that the upper bounds is, in fact, achieved by a non-adaptive strategy. In this case, being adaptive does not help.

Sometimes, instead of just determining whether a graph is connected or not, we want to know its connected components. It is obvious that both reachability and connectivity reduce in one step to finding connected components.

However, in the other direction we need in general to ask $\binom{n}{2}$ questions. We prove this fact for the connectivity oracle. $\mathscr{B}$ simply answers "no" to all questions: $\mathscr{A}$ cannot know the connected components before he queries all edge slots. If he does not query $\{(u, v)\}$ then he cannot tell whether the graph is empty or contains only one edge, (u, v).

3. Reducing Reachability to Connectivity

First we introduce our model and some definitions. We allow $\mathscr{A}$ to ask only questions of the form

(∗) For nonempty $X \subseteq E(V)$, is $G|_X$ connected?

By $C(n)$ we denote the maximum number of questions asked when $\mathscr{A}$ uses his best strategy, and the graphs in question have n nodes.

We first consider non-adaptive strategies. By $C_0(n)$ we denote the complexity of non-adaptive strategies in our model.

THEOREM 3.1.

$$C_0(n)=\begin{cases}\binom{n}{2} & \textit{for}\quad n\geqslant 6,\\ 2^{n-2} & \textit{for}\quad n=2,3,4,5.\end{cases}$$

Proof. The upper bound is obvious, since we can either query all edge slots, or query the vertex sets of all potential $1-n$ paths. In the first case we have $\binom{n}{2}$ questions, while in the second, all 2^{n-2} queries $E(\{1, n\}\cup Y)$ for $Y \subseteq \{2, ..., n-1\}$.

Now we prove the lower bound. Let Γ be the set of queries made by $\mathscr{A}$. For $2 \leqslant v < u \leqslant n-1$, let $\Gamma_{vu} \subseteq \Gamma$ be the set of queries Q such that $(v, u) \in Q \subseteq E(1, v, u, n)$. In order to distinguish between the case $E_G = \{(1, v), (u, n)\}$ and the case $E_G=\{(1, v), (v, u), (u, n)\}$, Γ_{vu} must be nonempty. Arbitrarily choose a $Q_{vu} \in \Gamma_{vu}$. Similarly, for $w \in \{2, ..., n-1\}$, let Γ_w be the set of queries Q such that $Q \subseteq E(1, w, n)$ and w is an endpoint of an edge slot in Q. Γ_w must be nonempty, in order to distinguish between the cases $E_G = \varnothing$ and $E_G = \{(1, w), (w, n)\}$. Arbitrarily choose a $Q_w \in \Gamma_w$. By a similar argument, we must have that $Q_0 = \{(1, n)\}$ must also be in Γ. The Γ_{vu}'s and Γ_w's are together pairwise disjoint and they do not contain Q_0. Hence, counting only Q_{vu}'s, Q_w's, and Q_0, we find that

$$|\Gamma| \geqslant \binom{n-2}{2} + (n-2) + 1 = \binom{n-1}{2} + 1.$$

This gives the lower bound for $n = 2$, 3, and 4.

We assume now that $n \geqslant 5$ and show that there must exist other queries besides Q_{vu}'s, Q_w's, and Q_0. Let σ be any cyclic permutation on $2, 3, ..., n-1$ and let $1 < v < n$. If $|\Gamma_v \cup \Gamma_{v,\sigma(v)}| \geqslant 3$, let $R_v \in \Gamma_v \cup \Gamma_{v,\sigma(v)} - \{Q_v, Q_{v,\sigma(v)}\}$. If instead $|\Gamma_v \cup \Gamma_{v,\sigma(v)}| \leqslant 2$, then $|\Gamma_v| = 1$ and $|\Gamma_{v,\sigma(v)}| = 1$. It is easy to verify that $|\Gamma_v| = 1$ implies that $(1, v)$, $(v, n) \in Q_v$. This, together with the condition $|\Gamma_{v,\sigma(v)}| = 1$, implies that $(1, v)$, $(v, \sigma(v)) \in Q_{v,\sigma(v)}$. Now, in order to distinguish between the cases

$$E_G = \{(1, \sigma^{-1}(v)), (\sigma^{-1}(v), v), (\sigma(v), n)\},$$

$$E_G = \{(1, \sigma^{-1}(v)), (\sigma^{-1}(v), v), (v, \sigma(v)), (\sigma(v), n)\},$$

we see that there must be a query $R_v \in \Gamma$ for which all three vertices $\sigma^{-1}(v)$, v, and $\sigma(v)$ are in $G|_{R_v}$ and $R_v \subseteq E(1, \sigma^{-1}(v), v, \sigma(v), n)$. Therefore, in both cases we have found a query R_v in Γ which has not yet been counted.

For $n = 5$, the R_v's give at least one additional query and the claimed lower bound follows. If $n \geqslant 6$, then all R_v's are distinct and thus

$$|\Gamma| \geqslant \left[\binom{n-1}{2} + 1\right] + n - 2 = \binom{n}{2}. \quad \blacksquare$$

Now we consider adaptive strategies. We show first that the upper bound can be slightly improved. Specifically, we exhibit a strategy for $\mathscr{A}$ that makes at most $\binom{n-2}{2} + n - 1$ queries. $\mathscr{A}$ first asks all questions $\{(u, v)\}$ for each $u, v \in \{2, ..., n-1\}$, $u < v$. After this $\mathscr{A}$ knows the connected components of the subgraph of G induced by vertices $2, ..., n-1$. For each of these connected components X, $\mathscr{A}$ now queries $E(X \cup \{1, n\})$. If there is a path of length at least two, then this path will now be found by $\mathscr{A}$. Finally, $\mathscr{A}$ queries $\{(1, n)\}$. Clearly, we have here at most $\binom{n-2}{2} + n - 1$ questions.

This upper bound can still be improved, as implied by the following theorem.

THEOREM 3.2. *For each $n \geqslant 2$,*

$$C(n) \leqslant \binom{n-2}{2} + \lfloor \log_2(n-1) \rfloor + 2.$$

Proof Sketch. $\mathcal{A}$'s algorithm consists of two phases. Let H be the subgraph of G induced by $V \backslash \{n\}$. First, by an appropriate choice of queries, $\mathcal{A}$ constructs a sequence of vertices

$$y_1 = 1, y_2, ..., y_{n-1} \in \{1, 2, ..., n-1\}$$

such that there is a k, $1 \leqslant k \leqslant n-1$, for which $\{y_1, y_2, ..., y_k\}$ is the connected component of H containing vertex $y_1 = 1$ and for $1 \leqslant i \leqslant k$, the subgraph induced by $\{y_1, y_2, ..., y_i\}$ is connected. (Given $y_1, ..., y_i$, to find y_{i+1} we ask $n-i-2$ questions $E(y_1, y_2, ..., y_i, x)$ for $n-i-2$ of the $n-i-1$ elements $x \in V \backslash \{n, y_1, ..., y_i\}$. Define y_{i+1} to be the first x for which the answer is "yes" or the remaining element if all answers are "no.") We can then find the connected component of vertex 1 in H by doing binary search through the sequence $y_1, ..., y_{n-1}$; with one more question we will know if n is connected to this connected component. ∎

In the case of adaptive strategies it is much harder to prove the lower bound. There are two problems here. First, there is a simple strategy which asks at most kn questions if the distance between 1 and n is at most k. Therefore, in order to "fool" $\mathcal{A}$ we cannot concentrate on short paths, as in the non-adaptive case. Second, given a sequence of queries and responses to those queries, it is NP-complete to determine if they are *consistent*, i.e., if there is an n-node graph consistent with the given responses to the specified queries. This is true even if all queries are of the form $E(U)$, for sets U satisfying $|U| \leqslant 4$. If all queries are of the form $E(U)$ for sets U of size at most three, then the problem can be solved in polynomial time. These facts are proven in the next section.

THEOREM 3.3. *$C(n)$ is $\Omega(n^2)$.*

Proof. Nodes s' and t' will be varying elements of V. Player $\mathcal{B}$ has partition (S, W, T) of V satisfying the following properties:

1. The subgraph induced by S is an $s-s'$ path, viewed as directed from s to s'. There are no edges between S and $V \backslash S$ except possibly some of those incident to s'.

2. The subgraph induced by T is a $t-t'$ path, viewed as directed from t to t'. There are no edges between T and $V \backslash T$ except possibly some of those incident to t'.

W will denote $V \backslash S \backslash T$. $\mathcal{B}$ hides from $\mathcal{A}$ all edges and nonedges in the subgraph induced by $W \cup \{s', t'\}$ (of course, $\mathcal{A}$ may learn something about $W \cup \{s', t'\}$ from the responses to his queries). All other edges and nonedges are revealed by $\mathcal{B}$ to $\mathcal{A}$.

Initially $S=\{s\}$, $T=\{t\}$, $s'=s$, $t'=t$, and $W=V\backslash\{s,t\}$. The idea is to extend S and T gradually, in such a way that each extension requires $\Omega(n)$ queries. In order to do this, we keep track of past queries which are of a certain form.

First, some terminology and notation. As S and T both induce paths, we speak of an *interval* of a path as a nonempty consecutive sequence of nodes in the path. A *suffix* of a path is either the empty set or an interval of the path that contains the final vertex.

To *append* a node to a path is to add than one node to the end of the path, adding the single edge between it and the previous endpoint. The concatenation of directed paths P and Q is denoted $P \circ Q$, while the reverse of P is denoted P^R.

Let α, β, and γ be constants such that $0<\alpha<\beta<\gamma<1$; their exact values will be determined later. By Γ we denote the set of queries that have already been made.

Let $P=(U,D)$ be a path in a graph. We say that a query Q *approximates* P if

$$D \subseteq Q \subseteq E(U).$$

In other words, (a) only edge slots between (possibly non-consecutive) vertices of P may occur in Q, and (b) every edge slot between consecutive vertices of P *is* in Q. We will have to deal especially carefully with queries approximating paths that consist of a suffix of S (or T) followed by a vertex x from W—if we answer "no" to such a query we cannot extend S to x without contradicting our response.

We introduce the following notation:

- Z_s is the set of all $x \in W$ such that for no nonempty suffix S_1 of S has any query approximating the path $S_1 \circ x$ been made.
- Similarly, Z_T is the set of all $x \in W$ such that for no nonempty suffix T_1 of T has any query approximating the path $T_1 \circ x$ been made.
- For $x \in W$, $Z_{S,x}$ is the set of all $y \in W\backslash\{x\}$ such that for no suffix S_1 of S (possibly empty) has any query approximating the path $S_1 \circ x \circ y$ been made.
- For $x \in W$, $Z_{T,x}$ is the set of all $y \in W\backslash\{x\}$ such that for no suffix T_1 of T (possibly empty) has any query approximating the path $T_1 \circ x \circ y$ been made.

The reader will see later that Z_S is the set of those x that can be appended to S without contradicting our previous responses, and $Z_{S,x}$ is the set which will become Z_S is x is appended to the path induced by S. Analogous statements hold for Z_T and $Z_{T,x}$. Here is $\mathscr{B}$'s algorithm.

$S := \{s\}$; $T := \{t\}$; $W := V \backslash S \backslash T$;
$s' := s$; $t' := t$;
repeat $\lceil \alpha n^2 \rceil$ **times**
begin
  Let $Q \subseteq E(V)$ be the current query;
  if Q approximates an interval of S or of T, **then** reply "yes"
  else begin
    reply "no" to the query Q;
    Adjust $Z_{S,x}$, $Z_{T,x}$ for all x and Z_S and Z_T, if necessary;
    if $|Z_S| \leqslant \beta n$ **then**
      if there is an $x \in Z_S$ s.t. $|Z_{S,x}| \geqslant \gamma n$ **then**
        begin
          Choose such a node x;
          Announce to $\mathscr{A}$ that (s', x) is the only edge between S and $V \backslash S$;
          $W := W \backslash \{x\}$; $S := S \circ x$; $s' := x$;
          Adjust $Z_{S,x}$, $Z_{T,x}$ for all x and Z_S and Z_T, if necessary
        end
      else HALT;
    if $|Z_T| \leqslant \beta n$ **then**
      if there is an $x \in Z_T$ s.t. $|Z_{T,x}| \geqslant \gamma n$ **then**
        begin
          Choose such a node x;
          Announce to $\mathscr{A}$ that (x, t') is the only edge between $V \backslash T$ and T;
          $W := W \backslash \{x\}$; $T := T \circ x$; $t' := x$;
          Adjust $Z_{S,x}$, $Z_{T,x}$ for all x and Z_S and Z_T, if necessary
        end
      else HALT
  end
end

The structure of the proof is as follows. First we assume that $\mathscr{A}$ continues to play as long as $\mathscr{B}$'s algorithm does not halt. With this assumption, we prove that by the time that $\mathscr{B}$'s algorithm halts, $\mathscr{A}$ must have made $\Omega(n^2)$ queries and still he does not have enough information to determine whether there is a path from s to t. Clearly, this implies that $\mathscr{A}$ could not possibly have stopped the game earlier, before $\mathscr{B}$ decides to do so.

Claim A. Whenever $\mathscr{B}$ halts, $|W| \geqslant \delta n - 3$, for $\delta = 1 - \alpha/(\gamma - \beta)$, if n is sufficiently large.

Note that one complete execution of the main loop might decrease $|Z_S|$ by one, or decrease $|Z_T|$ by one, or neither, but not both. The first queries will be known as $|Z_S|$*-decreasing queries*, the second, $|Z_T|$*-decreasing queries*. We show that strictly between any two extensions of S there must be at least $(\gamma-\beta)\,n\ |Z_S|$-decreasing queries.

After x is appended to S the new value of $|Z_S|$ is at least γn, because $|Z_{S,x}|$ was at least γn before x was appended to S, and the new value of Z_S is the old value of $Z_{S,x}$. The next extension of S is possible only when $|Z_S| \leqslant \beta n$. Thus, there must be at least $(\gamma-\beta)\,n\ |Z_S|$-decreasing queries between any two consecutive extensions of S. Analogous statements hold for T. Therefore, since initially $|S|=|T|=1$, at termination

$$(|S|+|T|-2)(\gamma-\beta)\,n \leqslant \lceil \alpha n^2 \rceil,$$

which easily implies Claim A.

Claim B. By the time when $\mathscr{B}$ halts, $\mathscr{A}$ has asked at least

$$\min\{\alpha n^2, \tfrac{1}{2}[\beta n-1][n(\delta-\gamma)-4]\}$$

queries. (Provided that $\delta>\gamma$, this is a quadratic function of n.)

Claim B is obvious if the main loop is executed all $\lceil \alpha n^2 \rceil$ times. The algorithm halts earlier only if at some time $|Z_S| \leqslant \beta n$ and $|Z_{S,x}| < \gamma n$ for all $x \in Z_S$, or $|Z_T| \leqslant \beta n$ and $|Z_{T,x}| < \gamma n$ for all $x \in Z_T$. By symmetry, assume that the first happens. Then $|Z_S| \geqslant \lceil \beta n - 1 \rceil$, because a query can decrease $|Z_S|$ by at most one. For each $x \in Z_S$, for each $y \in W \backslash Z_{S,x} \backslash \{x\}$, there is a suffix S_1 of S for which there was a query Q approximating the path $S_1 \circ x \circ y$. A query may contribute twice (if $x, y \in Z_S$, $y \in W \backslash Z_{S,x}$, and $x \in W \backslash Z_{S,y}$), but no more. Thus the number of queries is at least

$$\tfrac{1}{2} \sum_{x \in Z_S} (|W| - |Z_{S,x}| - 1) \geqslant \tfrac{1}{2}(\beta n - 1)(\delta n - 4 - \gamma n).$$

This completes the proof of Claim B.

Claim C. After $\mathscr{B}$ halts, $\mathscr{A}$ still does not have enough information to decide whether there is a path from s to t.

Obviously, there is always a way to complete the graph so that there is no path from s to t: simply omit all potential edges not already known to exist. We have to show that it is still possible to construct G in such a way that there is a path from s to t and G is consistent with the responses to the previous queries. G will consist of an $s-t$ path and the remaining vertices will be isolated.

This path is found as follows. Consider all potential paths

$$P_{xy} = S \circ x \circ y \circ T^R,$$

for $x \in Z_S$, $y \in Z_T$, $x \neq y$. For two distinct vertices $x \in Z_S$ and $y \in Z_T$, consider the path P_{xy}. If a query $Q \in \Gamma$ does not approximate an interval of P_{xy} then we have answered "no" to Q. If Q approximates an interval of S or T then we have answered "yes." Also, because of the choice of $x \in Z_S$ and $y \in Z_T$, there is no past query approximating an interval of P_{xy} containing exactly one of x and y. The only possibility that P_{xy} may not be consistent with Q occurs if Q approximates an interval of P_{xy} that contains x and y. Let us say that in this case Q *eliminates* P_{xy}. One query can eliminate two paths, P_{xy} and P_{yx}, but no more. But the number of paths P_{xy} is at least

$$\lceil \beta n - 1 \rceil \lceil \beta n - 2 \rceil,$$

because whenever and however the algorithm halts, $|Z_S|$ and $|Z_T|$ are at least $\lceil \beta n - 1 \rceil$.

Therefore, if $\beta^2/2 > \alpha$ and n is large enough, there are $x_0 \in Z_S$ and $y_0 \in Z_T$ with $x_0 \neq y_0$ such that $P_{x_0 y_0}$ is not eliminated by any query in Γ. The graph consisting of $P_{x_0 y_0}$ and the remaining points isolated is consistent with Γ. This completes the proof of Claim C.

Combining all the restrictions on our constants, we have

(a) $\delta = 1 - \alpha/(\gamma - \beta) > \gamma$,

(b) $0 < \beta < \gamma < 1$,

(c) $0 < \alpha < \beta^2/2$.

These conditions hold, for example, for $\alpha = \frac{1}{25}$, $\beta = \frac{1}{3}$, $\gamma = \frac{2}{3}$, and $\delta = \frac{22}{25}$. The proof of Theorem 3.3 is now complete. ∎

4. On Determining Consistency

In this section we consider the problem of checking the consistency of a set of responses to queries about connectivity of induced subgraphs. The problem is

Connectivity Consistence. COCO: Given $V = \{1, 2, ..., n\}$ and two sets Γ_Y, Γ_N of queries $Q \subseteq E(V)$, is there a graph $G = (V, E)$ such that

(a) each subgraph $G|_Q$ for $Q \in \Gamma_Y$ is connected, and

(b) each subgraph $G|_Q$ for $Q \in \Gamma_N$ is not connected.

Theorem 4.1. *The problem COCO is NP-complete, even if all queries have the form $Q = E(Z)$ for sets Z of size at most four.*

Proof. Clearly COCO $\in$ NP. We reduce 3-SAT to COCO. Let $x_1, ..., x_n$ be the variables and let $C = C_1 \wedge \cdots \wedge C_k$ be the instance of 3-SAT, with

$|C_i| = 3$ for all i. We transform this instance into an instance of COCO. We take

$$V = \{x_1, \bar{x}_1, ..., x_n, \bar{x}_n, z\},$$

where z is an additional vertex. Γ_Y contains the queries

1. $\{(x_i, x_j)\}$, $\{(x_i, \bar{x}_j)\}$, $\{(\bar{x}_i, x_j)\}$, $\{(\bar{x}_i, \bar{x}_j)\}$, for all $1 \leqslant i < j \leqslant n$.
2. $E(C_i \cup \{z\})$, for all $i = 1, ..., k$.

Γ_N contains the queries

1. $\{(x_i, \bar{x}_i)\}$, for all $i = 1, ..., n$,
2. $E(x_i, \bar{x}_i, z)$, for all $i = 1, ..., n$.

Note that at most one edge of (x_i, z), $(\bar{x}_i, z)$ may be in G. G corresponds to a satisfying assignment f such that (x_i, z) is an edge in G if and only if $f(x_i)$ is "true." It is quite easy to see that this polynomial-time transformation has the required properties. ∎

The theorem below shows that the results above cannot be improved to queries of size three, unless P = NP.

THEOREM 4.2. (Chengdian Lin). *If all queries are of the form $E(Z)$ for a set Z of size at most 3, then COCO can be solved in polynomial time.*

Proof. We show that in this case the problem can be reduced to 2-SAT. Regard each edge slot e as a boolean variable, and if f is a truth assignment, $f(e) = 1$ means that e is an edge, and $f(e) = 0$ means that e is not an edge. For vertices $x, y \in V$, $x < y$, let e_{xy} be the boolean variable for slot (x, y). We construct a set $\mathscr{C}$ of clauses as follows. Suppose $Q = E(x, y, z)$, $x < y < z$. Observe that the subgraph induced by Q is connected if it contains at least two edges, and is disconnected if it contains at least two nonedges. If $Q \in \Gamma_Y$, then add to $\mathscr{C}$ the clauses $\{e_{xy} \vee e_{xz}\}$, $\{e_{xy} \vee e_{yz}\}$, and $\{e_{xz} \vee e_{yz}\}$. If $Q \in \Gamma_N$, then add to $\mathscr{C}$ the clauses $\{\bar{e}_{xy} \vee \bar{e}_{xz}\}$, $\{\bar{e}_{xy} \vee \bar{e}_{yz}\}$, and $\{\bar{e}_{xz} \vee \bar{e}_{yz}\}$. If $Q = \{(x, y)\}$ (for $x < y$) is in Γ_Y or Γ_N, simple add clause $\{e_{xy}\}$ or $\{\bar{e}_{xy}\}$, respectively. It is easy to see that this set of clauses has a satisfying assignment if and only if there is a graph G consistent with the responses to the queries. ∎

ACKNOWLEDGMENTS

We thank Chengdian Lin for allowing us to include Theorem 4.2 in the paper. We also thank J. Naor, M. Yung, and G. Turán for helpful discussions.

RECEIVED August 15, 1989; FINAL MANUSCRIPT RECEIVED November 7, 1989

REFERENCES

AGGARWAL, A., COPPERSMITH, D., AND KLEITMAN, D. (1989), A generalized model for understanding evasiveness, *Inform. Process. Lett.* **30**, 205–208.

EVEN, S., ITAI, A., AND SHAMIR, A. (1976), On the complexity of timetable and multicommodity flow problems, *SIAM. J. Comput.* **5**, 691–703.

HAJNAL, A., MAASS, W., AND TURÁN, G. (1988), On the communication complexity of graph properties, *in* "Proceedings, 20th ACM STOC'88," pp. 186–191.

KARCHMER, M., AND WIDGERSON, A. (1988), Monotone circuits for connectivity require super-logarithmic depth, *in* "Proceedings, 20th ACM STOC'88," pp. 539–550.

LOVÁSZ, L., AND SAKS, M. (1988), Lattices, Möbius functions and communication complexity, *in* "Proceedings, 29th IEEE FOCS'88," pp. 81–90.

RAZ, R., AND WIDGERSON, A. (1989), An $\Omega(\log^2 n)$ monotone depth lower bound for connectivity, manuscript.